Husky Navigator (Sillicon Valley Campus)

Ren Huang (huang.r1@northeastern.edu)
https://github.com/Ren-97/Husky_Navigator.git

1. Executive Summary

Husky Navigator is a Retrieval-Augmented Generation (RAG) system designed to provide students, faculty, and staff at Northeastern University's Silicon Valley campus with instant access to information about academic programs, courses, faculty, and important dates. By leveraging LangChain with Meta's open-source Llama 3 LLM and implementing a user-friendly Streamlit interface, this project creates an intelligent conversational assistant that overcomes the knowledge cutoff limitations of traditional LLMs while providing reliable, campus-specific information. This preloaded information approach allows the system to answer questions about the 2025 academic year.

2. Motivation

University information systems are often fragmented across multiple platforms, making it difficult for students to quickly access crucial information about course offerings, program requirements, faculty information, and academic deadlines. Traditional large language models (LLMs) have a limited ability to provide accurate, institution-specific information due to knowledge cutoff limitations and a lack of domain-specific training.

The Husky Navigator addresses these challenges by using a RAG approach that combines the conversational abilities of Llama 3 with retrieval from a curated collection of university documents. This allows the system to provide specific and accurate information about the Silicon Valley campus.

The project benefits:

- Students & Prospective Applicants: Access to current course descriptions, faculty information, academic calendars, and degree requirements.
- Advisors & Counselors: Fast retrieval of academic regulations and program details.
- Faculty & Administrators: Streamlined access to campus-specific information.

3. Background and Datasets

The Husky Navigator system leverages a diverse collection of Northeastern University Silicon Valley campus documents as its foundational knowledge base for information retrieval.

- **Academic Calendar (2025)**: University schedule of important dates, deadlines, and academic periods.
- **Course Descriptions**: Comprehensive information about course content, prerequisites, and learning objectives.

- **Graduate Catalog (Degree Requirements)**: Formal documentation of degree program structures, required courses, and graduation criteria.
- **Class Schedules (Summer & Fall 2025)**: Manually collected and formatted information about upcoming course offerings, including instructors, meeting times, and room assignments.
- **Faculty & Staff Information**: Manually compiled profiles containing faculty expertise, contact information, and academic backgrounds to facilitate connections between students and instructors.

This hybrid approach of combining official university documentation with manually collected current information ensures comprehensive coverage while maintaining accuracy. The document collection was structured with consistent metadata to optimize retrieval precision for different query types, enabling the system to provide specific answers about course offerings, faculty information, degree requirements, and important academic dates.

4. Modeling Methodology

Husky Navigator implements a sophisticated multi-stage process for information retrieval and generation.

1. Data Processing Pipeline:
   - ➢ Document loading using LangChain's specialized loaders (PyMuPDFLoader, CSVLoader).
   - ➢ Custom metadata enrichment to categorize information .
   - ➢ Type-specific text splitting strategies with customized chunk sizes and overlap.
   - ➢ Embedding generation using the nomic-embed-text model.
   - ➢ Storage in a Chroma vector database for efficient similarity search.

2. Tool-Based Architecture:
   - ➢ Six specialized tools for different query types (course_search, faculty_search, academic_calendar, degree_requirements, course_schedule, northeastern_knowledge_base).
   - ➢ LLM-based tool selection for optimal query routing
   - ➢ Fallback RAG mechanism for queries that don't fit standard categories

3. Enhanced RAG Chain with Multi-Step Reasoning:
   - ➢ Query reformulation to improve retrieval effectiveness
   - ➢ Context-aware reasoning with the retrieved information
   - ➢ Structured response generation with appropriate formatting and tone

4. Streamlit User Interface:
   - ➢ Chat-based interface
   - ➢ Visual indicators of which tool was used for each response
   - ➢ Toggle for memory mode to enable/disable conversation context
   - ➢ Summary mode option for shorter responses
   - ➢ Reset functionality to clear conversation history

The system is containerized using Docker with an entrypoint script that initializes the Ollama server and loads the required models before starting the Streamlit application.

Repository: https://github.com/Ren-97/Husky_Navigator/blob/main/README.md

5. Results and Evaluation

The Husky Navigator system received rigorous evaluation through both automated and manual testing methodologies. The evaluation employed LangChain's QA evaluation framework, utilizing QAGenerateChain for test set creation from document samples and QAEvalChain for LLM-assisted assessment. This was supplemented with manually created test cases that represent common student inquiries, as the automated QAGenerateChain does not always capture the types of questions students typically ask about courses, schedules, and faculty.

Assessment focused on tool selection accuracy, measuring the system's ability to route queries to appropriate specialized tools and response quality metrics including factual correctness, coherence, and relevance. The multi-tool architecture with LLM-based query routing demonstrated superior performance compared to a general RAG implementation, particularly for queries requiring information from specified documents.

It's important to note that while QAGenerateChain and QAEvalChain provided valuable reference points for evaluation, they cannot determine which specialized tool should be used for a given query. Tool selection requires LLM-based reasoning about query intent that goes beyond the capabilities of standard QA evaluation frameworks. This highlights the value of the system's custom LLM-based tool selection mechanism, which analyzes queries to route them to the most appropriate specialized retrieval path.

The Husky Navigator system implements a distinctive approach compared to traditional RAG systems. Since users require precise answers to their university-related questions, the system doesn't need to refine documents during retrieval. Instead, it routes each query to the most appropriate specialized tool and document collection. This domain-specific tool selection provides more targeted and accurate information retrieval. When the specialized approach fails to resolve a query, the system gracefully transitions to a fallback mechanism that performs general retrieval across all documents.

The system incorporates an intelligent fallback mechanism that activates when specialized tools encounter exceptions or fail to retrieve relevant information. Rather than returning an error to the user, this approach seamlessly transitions to a comprehensive RAG chain that searches across all document collections, ensuring the user still receives a helpful response. This architectural resilience maintains system reliability even when confronted with unexpected query types or processing issues.

6. Conclusions

Through the development of the Husky Navigator system, it demonstrated that domain-specific retrieval systems can significantly outperform general-purpose approaches when tailored to the unique characteristics of university data structures and common query patterns.

The tool-based architecture proved particularly effective for university information retrieval, as academic queries naturally fall into distinct categories that benefit from specialized processing. The implementation of document-specific chunking strategies yielded significant improvements in retrieval precision, confirming that content structure should drive chunking decisions rather than applying uniform strategies across all document types.

It is important to note that due to website restrictions, it was not possible to directly scrape information from Northeastern University's website, nor to obtain streaming data. Therefore, the system uses static data containing information up to the end of 2025. These data limitations highlight the importance of addressing information acquisition challenges in real-world application scenarios, while also providing clear direction for future system upgrades.

Despite the system's overall effectiveness, several challenges remain to be addressed. The LLM still occasionally makes errors when determining which tool to use for a given query. This could be improved by further refining the prompts to help the LLM more accurately identify the appropriate tool based on query content, or by exploring alternative approaches to tool selection. Another limitation is that the system cannot simultaneously process multiple questions in a single conversation turn, which also requires enhancement to improve the user experience.

The Husky Navigator shows how RAG system can make university information easier to retrieve. By using specialized tools for different types of questions, it helps students and faculty quickly get the information they need. Although this version uses preloaded data, it lays the groundwork for future versions that could connect to live university systems.